

Project 4 – Design and Reflection Document

Problem:

Project 4, Fantasy Combat Tournament, is part 2 of an on-going assignment that incorporates concepts such as inheritance/polymorphism, linked lists, stacks, and queues. This program uses the Character classes we built in Project 3. Instead of 1v1 gameplay, this program allows the user to select as many Characters as they'd like for 2 lineups. The lineups are designed in queue-like linked lists. The teams fight to the death, with each loser being tossed into a stack-like loser pile.

Requirements:

- a) Display menu to select characters.
- b) Combat includes: Player 1 Attacks, Player 2 Defends; Player 2 Attacks, Player 1 Defends; Calculate damage and update stats.
- c) Continue game until one of the characters is at 0 or less strength points.
- d) After each round, the loser goes into the loser pile. Winner recovers a random percentage of SP, and is placed to the back of the lineup.
- e) The tournament ends when one of the teams is out of fighters, or when there is a tie (both teams out of players, or both team's remaining players kill each other in the same round).
- f) At the end of the game, asks player if they want to view the loser pile. Afterwards, asks if they'd like to play again or exit.

Program Design:

See attached flowchart.

This project involves the use of custom containers that behave like linked lists. Both lineup and loser pile linked-lists are non-circular, which made implementation a bit easier.

I reused my code from previous labs for linked lists, as well as for the initial Fantasy Combat game.

The Menu class and Die class are reused from previous assignments, and they are tailored to this program. The Menu class handles input validation as well as the main playthrough of the game. I chose to put play() inside Menu for easy access to character selection choices and validation functions. It also keeps my Main cleaner looking.

Test Plan:

See attached PDF.

Reflection:

Phew! I was tight on time with this project due to job and personal commitments, so unfortunately I was not able to give it my all. I was able to figure out the basic mechanics of the tournament. I thought this would be an easy implementation of linked lists, but having to keep stack's LIFO nature in mind somehow was more difficult than expected.

It is really interesting to see how the implementation changes when a team-organization is involved. I might've underestimated some of the details associated with moving from a 1v1 to a n vs. m type arena battle. One of the biggest issues that I've come across is losing track of where my memory went! I was unable to completely free up all the memory since I split it up between 5 characters x 2 teams + a loser pile.

Although we are heading towards the end of the class, I feel like I still have a lot of studying to do when it comes to pointers and memory management. I am getting less segmentation faults now, but am still having trouble tracking down leaks. This is something I will focus on over Spring break.

I like that we had a little break between Project 3 and 4, but I found myself struggling to remember how I implemented Project 3. There are a few areas, such as working with destructors, that I did not pay attention to in the first project. It is now haunting me in this second half!

It's really great to see my program evolve. I can't wait to dig into the Final Project!